SP8 Programmers

User's Guide

深圳硕飞科技有限公司

SHENZHEN SOFI TECHNOLOGY CO. , LTD.

TEL: 0755 - 8486 7757

FAX: 0755 - 8486 7941

WEB: www.sofi-tech.com

Publication Release Date: August 2011

Revision A1

Contents

Chapter 1. Introduction into Products ····· 3
Performance and Characters ····· 3
Correlation Table of SP8 Performance ····· 4
Hardware of SP8 Programmers ····· 5
Chapter 2. Software Installation ····· 6
Install FlyPRO ····· 6
Install USB Driver ····· 10
Chapter 3. Rapid Use ····· 13
Operation Procedure ····· 13
Place Chips ····· 14
ISP Program ····· 15
Offline program ····· 17
Chapter 4. Software Performance ····· 18
Introduction into Software ····· 18
Choose Chips ····· 19
Load Files ····· 20
Save Files ····· 21
Operation Options ····· 22
Automatic Serial Number ····· 23
Buffer Compile ····· 25
Chip Configuration ····· 27
Chip Information ····· 28
Pin Detection ····· 29
Download Offline Data ····· 30
Examine Offline Data ····· 31
Appendix one, Frequently Asked Questions ····· 32
Appendix two, Common Prompt Messages in Programming ····· 33

Chapter 1. Introduction into Products

Performance and Characters

- USB2.0 communication interfaces (real USB interfaces, not USB to serial ports or simulated USB ports)
- Exquisite shells customized with opening moulds, small (size: 103x71x23mm), and convenient to use and carry
- Built-in 32bit high-speed processors, designed specially for serial memories, achieving high-speed and appropriate programming
- Supports nearly all 93/24/25/BR90 memories, and supports more chips through software upgrade
- Supports ISP program. On-board patch chips can be directly connected to be programmed
- Standard 40Pin ZIF socket, applicable to wide/narrow chips and general adapters
- Supports contact detection for pins, thus improving the reliability of programming
- Perfect over-current and ESD protection, and protection against accidental damages caused by misplaced or destroyed chips to programmers and computer USB ports
- With mass production program, SP8 automatically detects the position of chips and then starts program operation
- Supports mass asynchronous program by connecting several programmers to one computer. Plug and write. No synchronous waiting. Top speed and high efficiency
- You can choose offline program: built-in 128Mbit data memories achieve different kinds of field programming without connection to a computer, and LED and buzzer present voice prompt for success or failure. It operates easy and you can perform mass program
- Supports low-voltage (3.3V) and 5V chips
- Provides self-detection for devices
- Supports WinXP, VISTA, Win7(32bit/64bit)

Note:

1. These functions are designed according to specific types, and only SP8-F has all these functions. For details, please refer to the attached correlation table for types.
2. ISP mode does not support offline program.

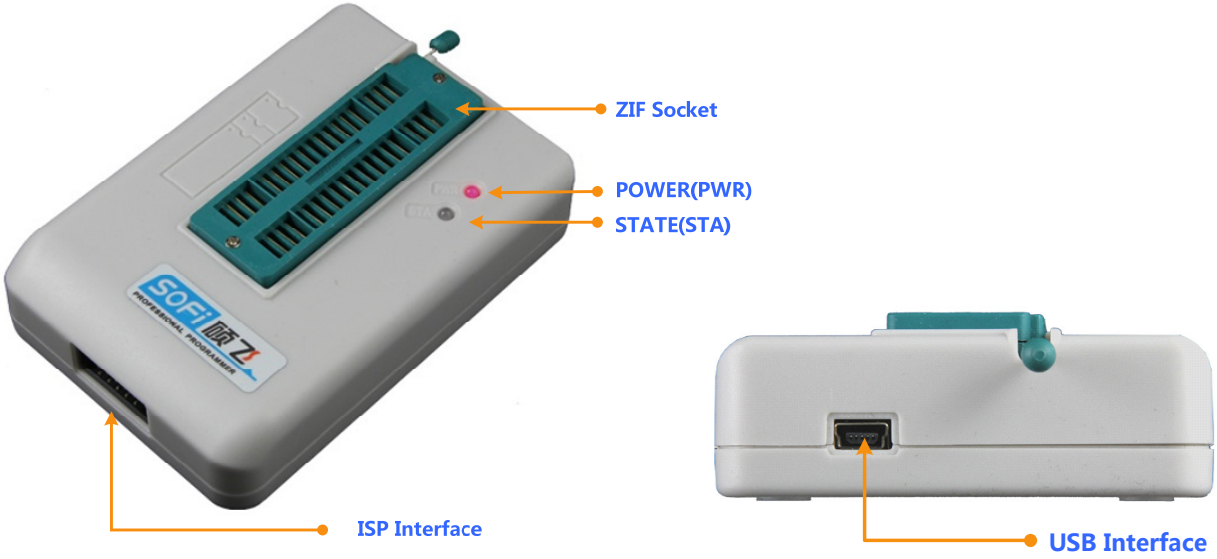
Correlation Table of SP8 Performance

SP8 Series includes three types, and their comparisons of functions are as follows:

Types	SP8-A (popular edition)	SP8-B (mass production edition)	SP8-F (offline edition)
ISP supportive	√	√	√
Offline mass production	×	×	√
Online mass production	×	√	√
Socket types	40pin	40pin	40pin
Buzzer prompts	×	×	√(offline mode)
Pin contact detection	√	√	√
Protection against over-current	√	√	√
Automatic Serial Number	×	√	√
Supportive chip number (Note 1)	4253	4585	4588
Chip configuration	×	√	√
Application targets	Engineering Program/Home appliance and digit maintenance	Small amount program/Engineering Program	Mass production program

Note 1: The number of supportive chips is based on the newest software, and is to be upgraded. The table is up to Aug. 2, 2011.

Hardware of SP8 programmers



- **Socket**

  Used to place the chips in (including adapters)

- **Power indicator light (PWR)**

  Indicates status of power supply. PWR keeps red in normal supply (SP8 programmers adopt power supply through USB ports); it flashes red if the programmer has found short circuit or over-current.

- **Status indicator light (STA)**

  STA Declaration for programmers (online mode):

Status of STA	Declaration
Orange	Programming chips
Green	Succeeded in programming
Red	Failed in programming

STA Declaration for offline program (only for SP8-F):

Status of STA	Declaration
Flashes green	Waiting for chips to be put in
Orange	Programming chips
Green	Succeeded in programming, waiting for the chip to be removed
Red	Failed in programming, waiting for the chip to be removed

- **ISP interface**

  Used to connect target chips in ISP program mode. Choose the chip with suffix [ISP], and you will see the specific connection diagram in Chip Information.

- **USB interface**

  Used to connect the computer's USB port for communication, and supply power to the programmer

  Used to connect special power adapters in offline mode (only for SP8-F)

Chapter 2. Software Installation

Install FlyPRO

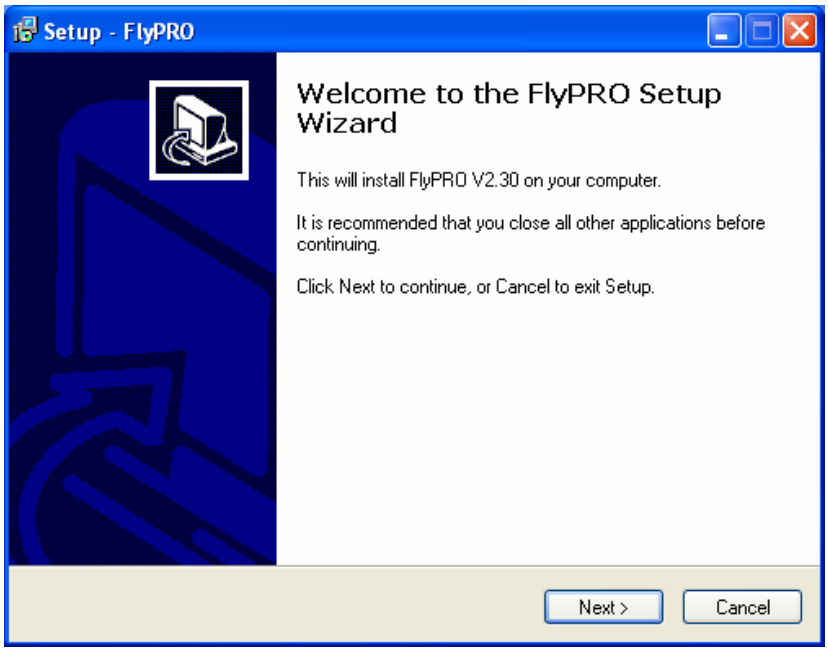
- **Get the software**

  The installation program is included in the product CD, under the directory SETUP.

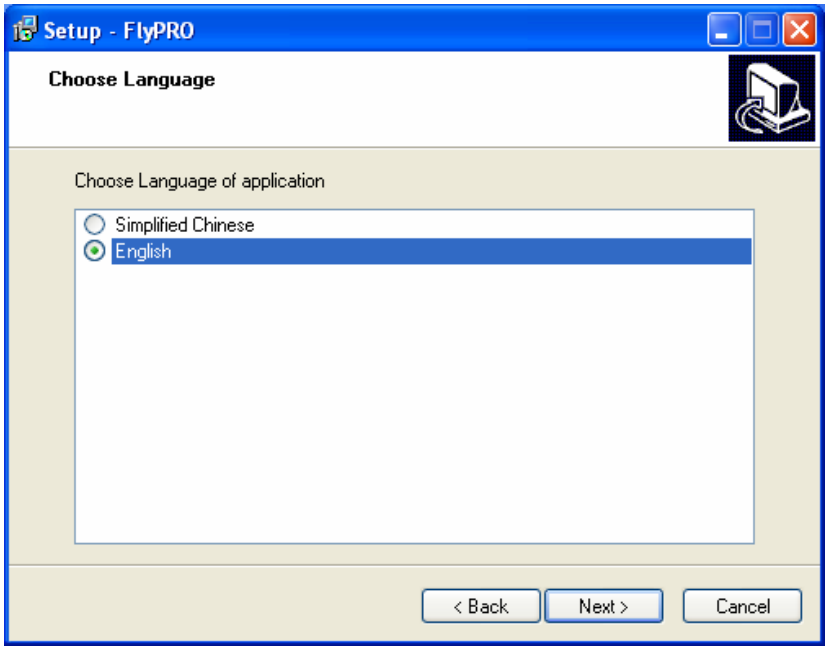
  And it can also be downloaded from SOFI's web (We recommend this, so you can get the newest version).

  Link for the newest software: http://www.sofi-tech.com/upload/Software/FlyPRO_Setup.rar

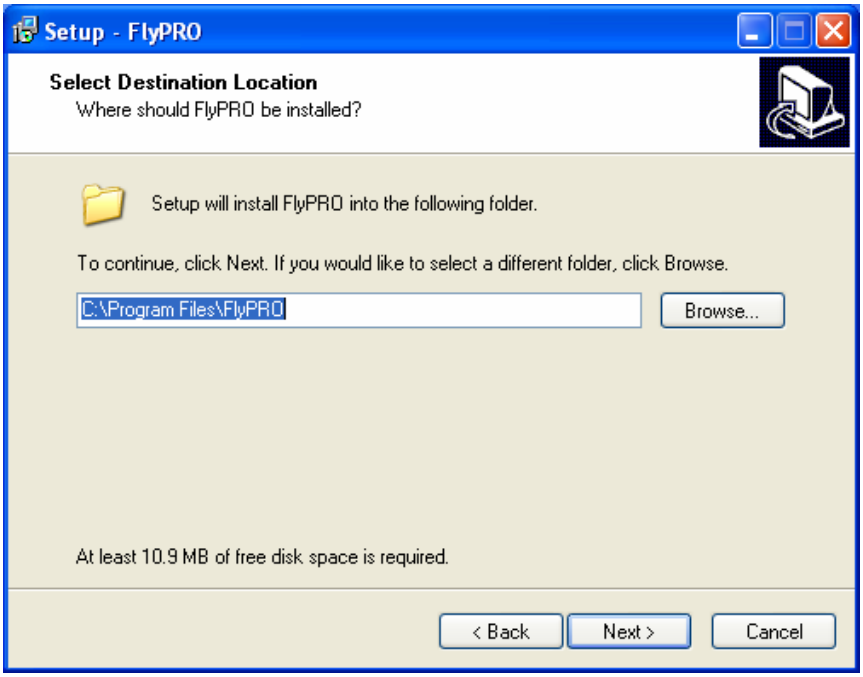
- **Start to install**



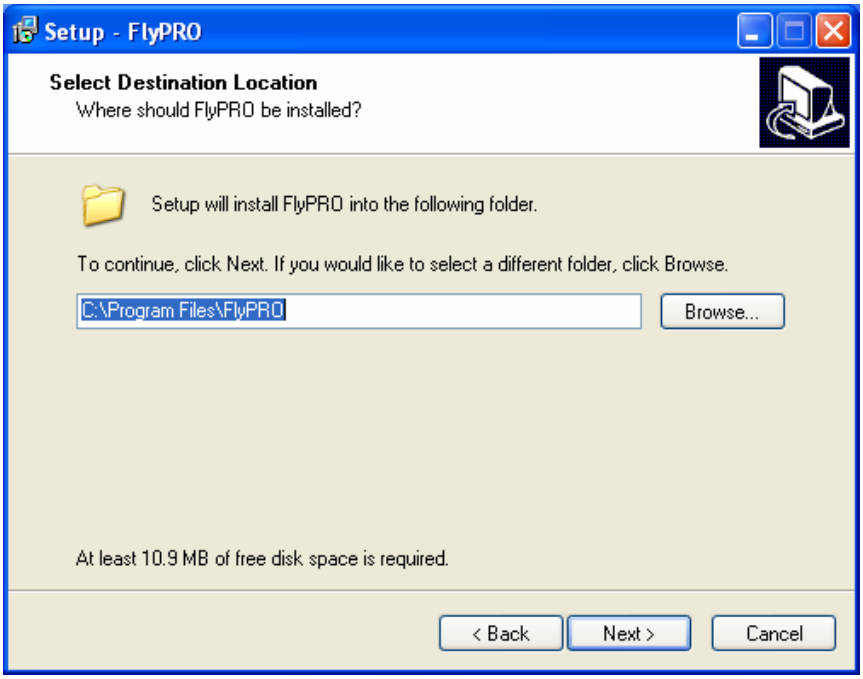
- **Choose Language**



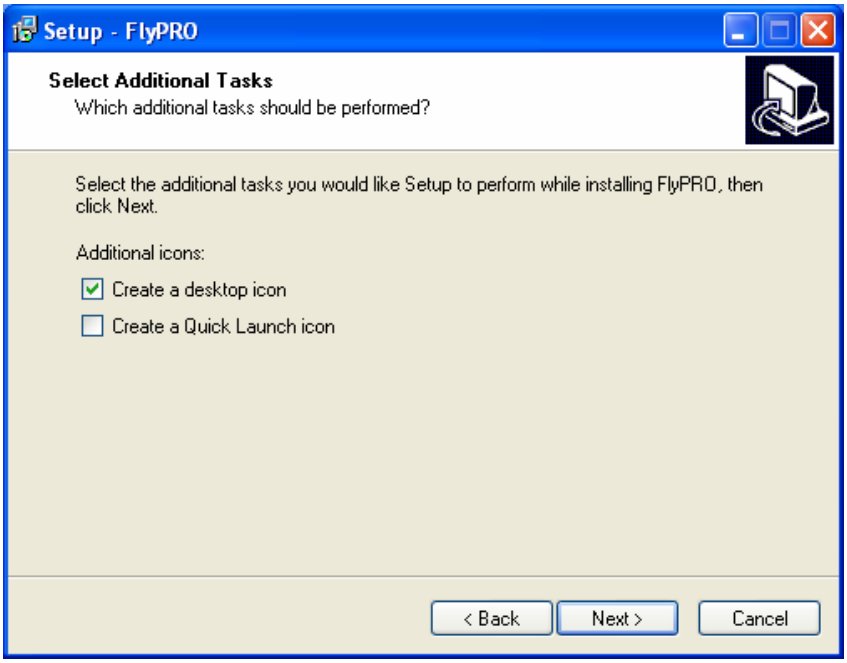
- Choose the directory to install



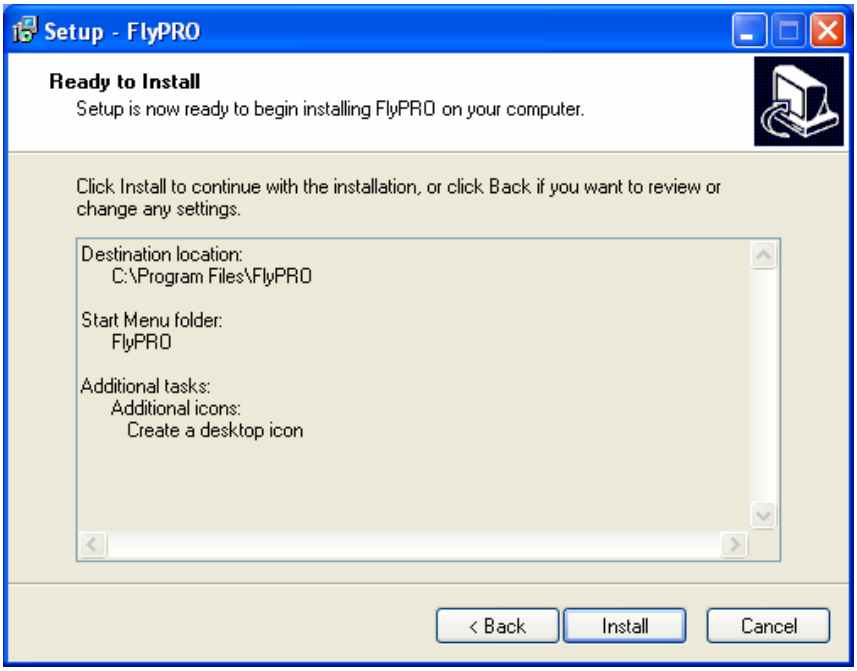
- Choose the folder for Start Menu



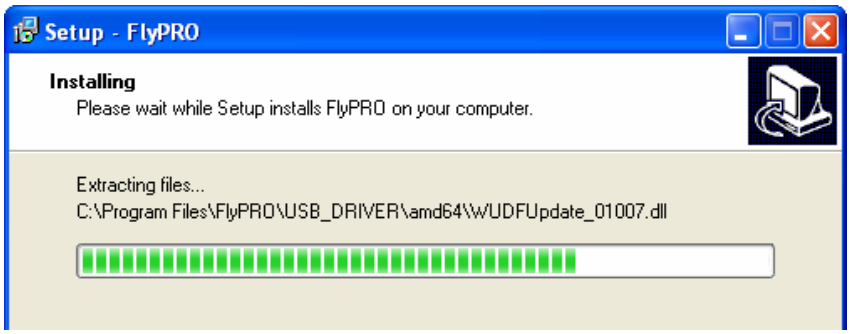
- Set whether to create shortcut on desktop



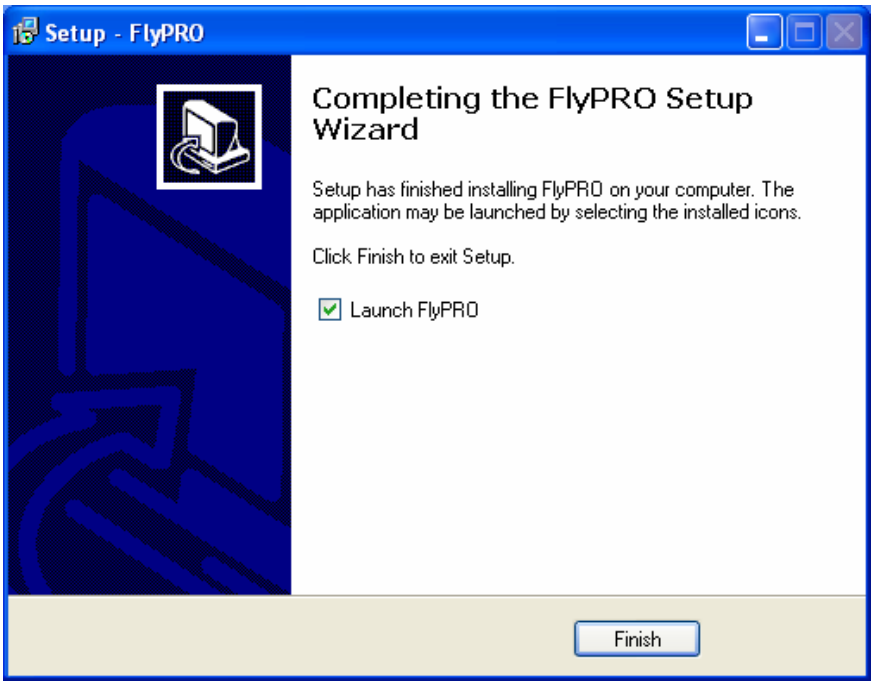
- Prepared to install



- Installing



- Completed

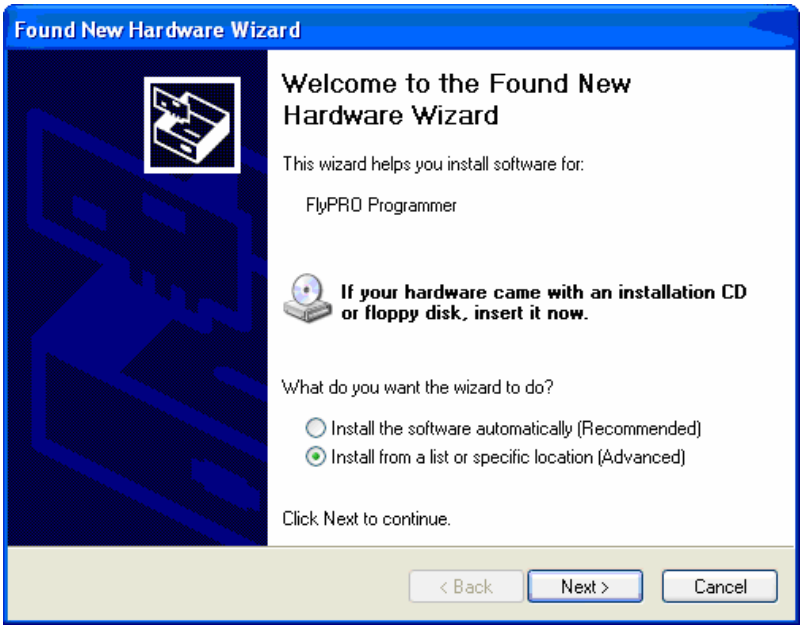

Setup - FlyPRO
Completing the FlyPRO Setup Wizard
Setup has finished installing FlyPRO on your computer. The application may be launched by selecting the installed icons.
Click Finish to exit Setup.
Launch FlyPRO
Finish

Install USB driver

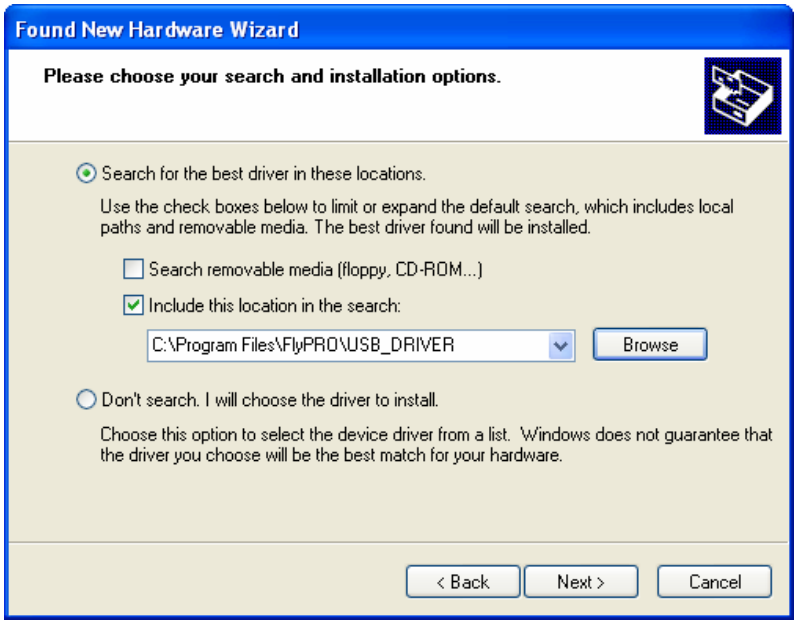
- The first time to connect the programmer to a computer, the computer automatically prompts "Found New Hardware" and requires to install driver.

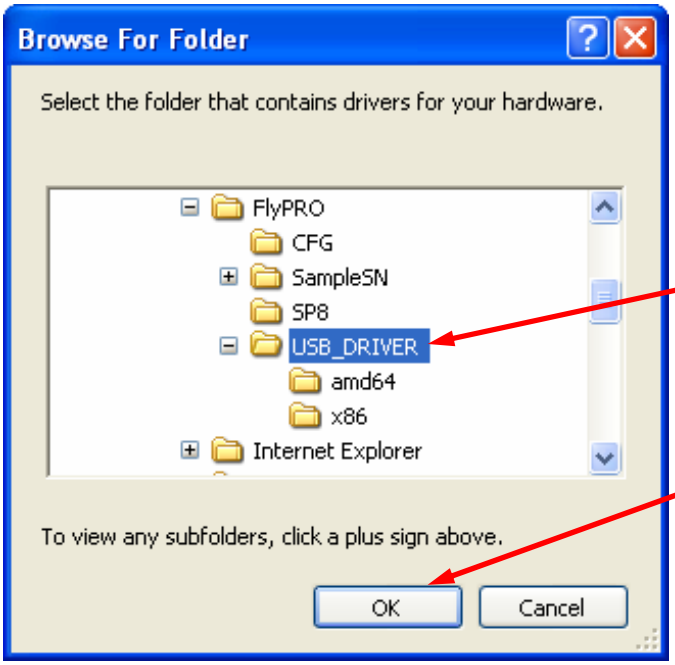
- Choose "No, not this time", and click "next"

- Choose "Install From a list or specify location (advanced)", and click "Next"



- Set according to the picture below: click "Browse", choose the folder "USB_DRIVER" under FlyPRO installation directory, and click "Next"



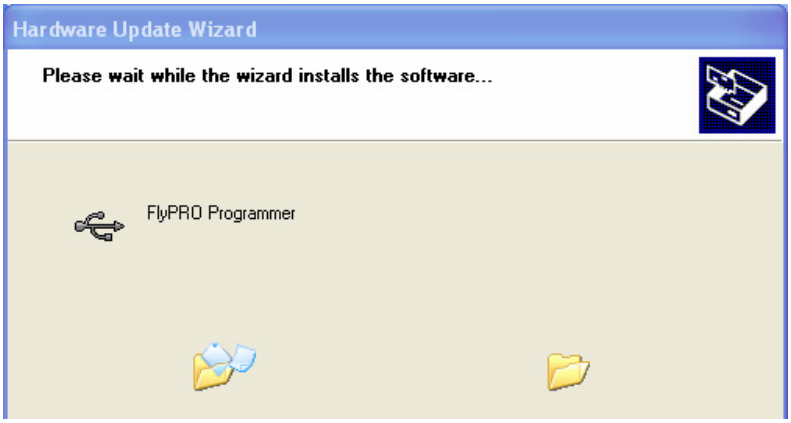


Click "Browse" in the previous picture, and select the folder USB_DRIVER under the installation directory of FlyPRO.

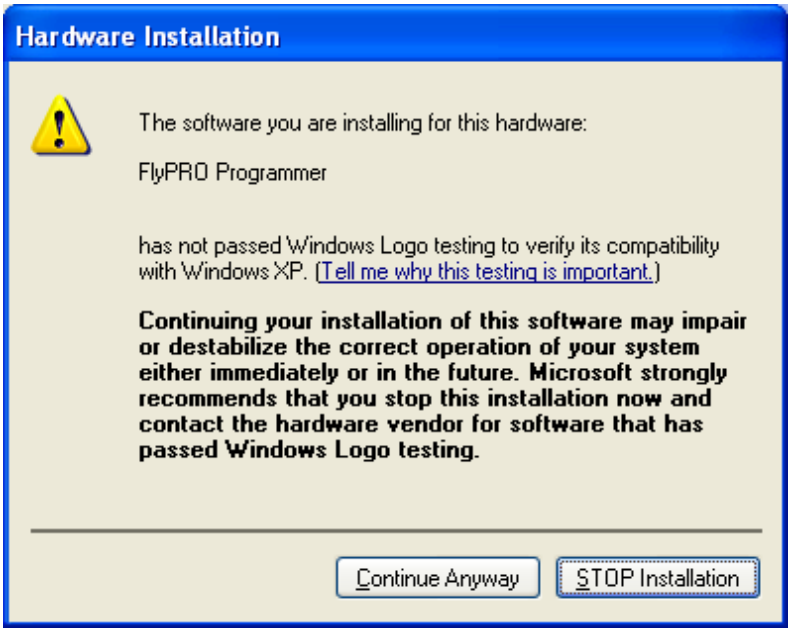
The folder selected must be USB_DRIVER , not its sub-folder (such as x86 or amd64).

Finally, click "OK", and go back to the previous picture. If this button is gray, there must be something wrong with folder you selected.

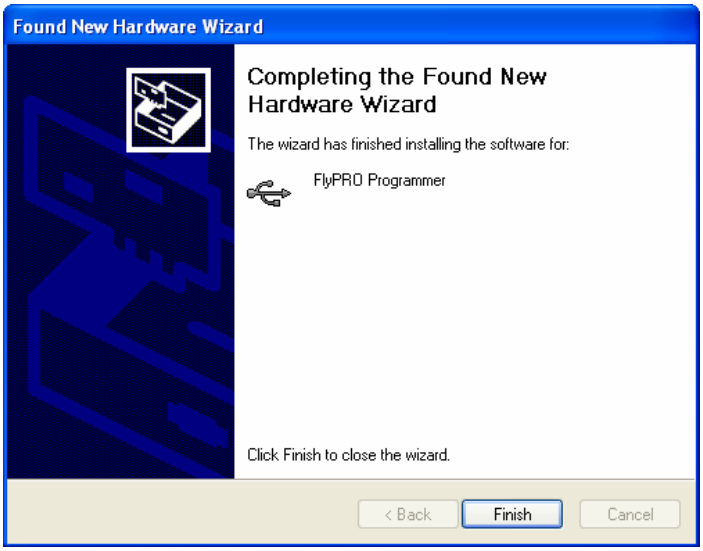
- Starts to install USB driver



- In installing, Windows prompts that the software hasn't passed Windows Micro scale test, so choose "Still go on"



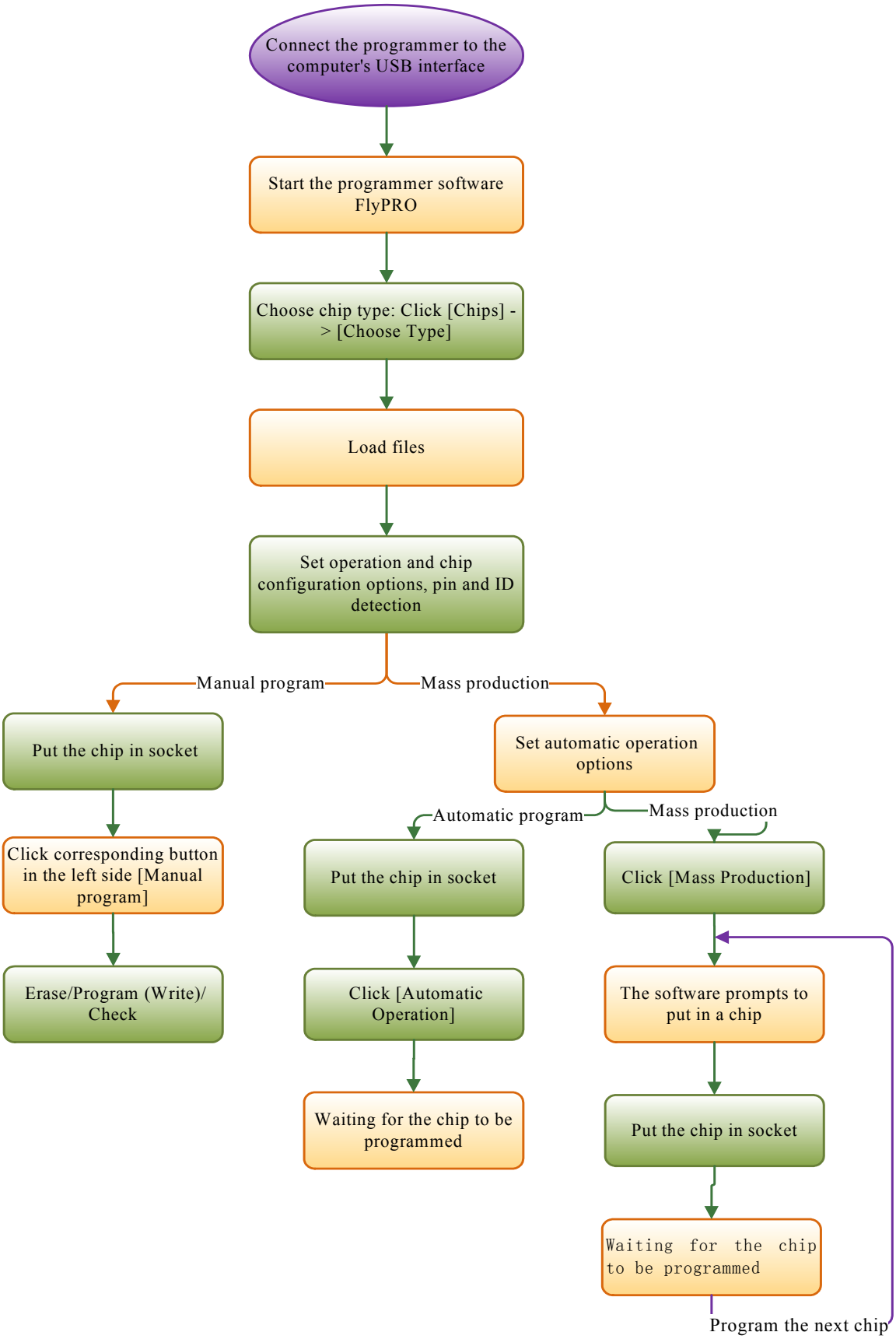
- USB driver installed



Chapter 3. Rapid use

Operation Procedure

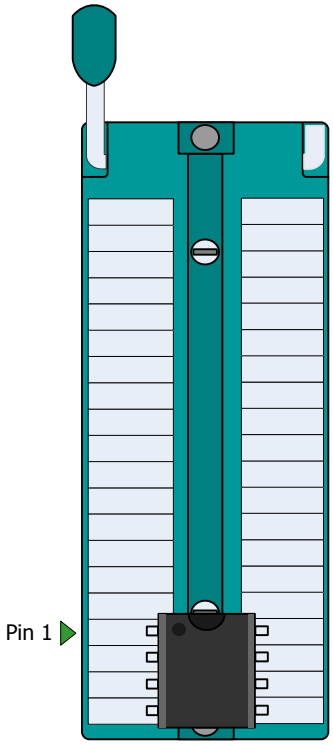
Declaration: To use the programmers, you should install FlyPRO and USB drivers. For details, please refer to Chapter 2.



Put the chips

SP8 adopts 40pin sockets for wide/narrow dual uses which support inline chips and adapters with 8 to 40 pins.

Placement for common chips:



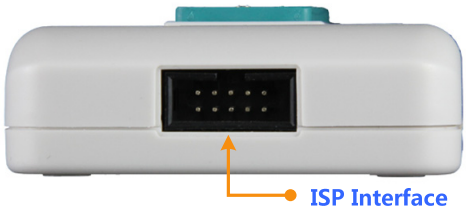
To see placement for other chips (including adapter information), please open Chip Information in FlyPRO

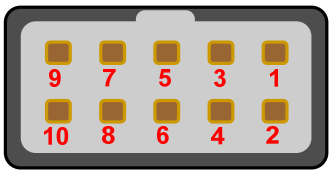
ISP program

ISP is short for In System Program, also called online program. In ISP mode, chip reading and writing can be operated simply by connecting several information lines to corresponding pins in the on-board chips, without chip unsoldering.

ISP interface

SP8 programmers provide an extra ISP interface. See the pictures below:





ISP Interface definiens

ISP connecting line

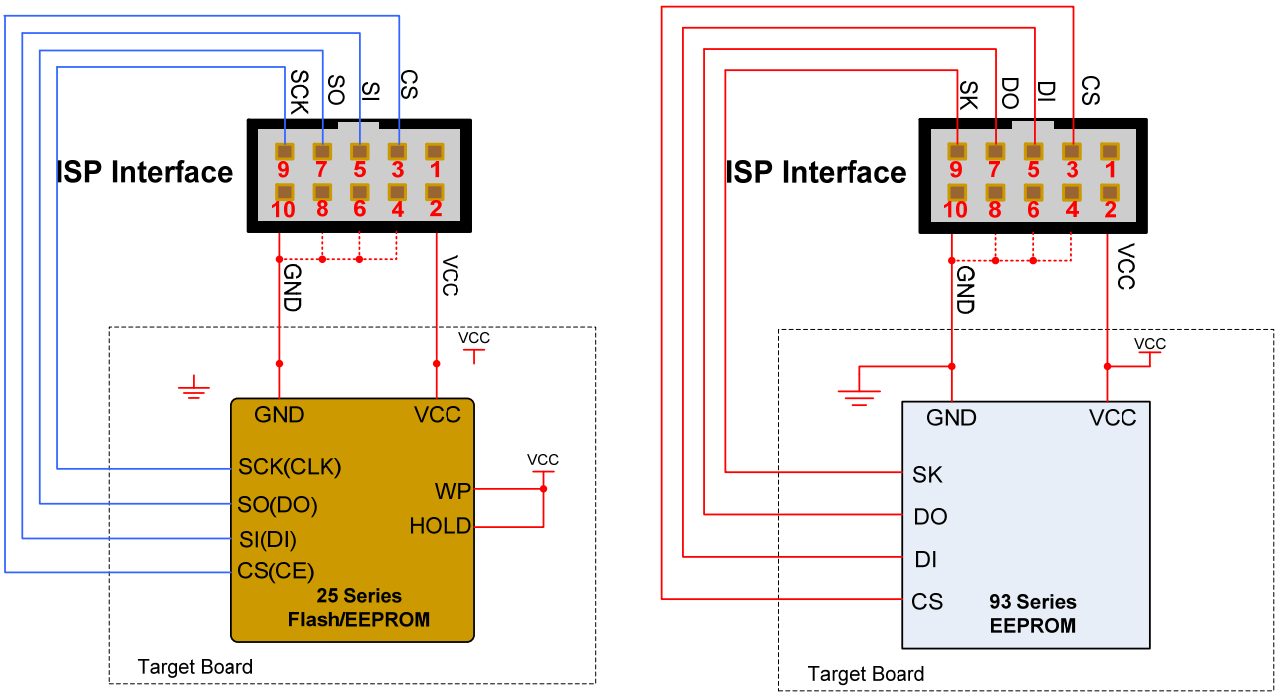
ISP connecting line is one 10-color ribbon line, with a 5x2 standard plug on one end to connect to ISP interface, and 10 DuPont heads on the other end to connect to corresponding pins on the target board.

Correlations between colors of connecting lines and serial numbers of pins are:

Color	Serial number of pins	Color	Serial number of pins
Brown	1	Blue	6
Red	2	Purple	7
Orange (or pink)	3	Grey	8
Yellow	4	White	9
Green	5	Black	10

Connect target chips

Connecting pictures for common 93 and 25 EEPROM are:



There are different connecting ways for different chips. For details, please open Chip Information in FlyPRO.

Power supply for ISP chips

In ISP program, there are two ways of power supply for target chips. You can set the supply way in the operation options of FlyPRO.

- **Power supply by programmers**

  Power supplied to target boards by programmers, with voltage 3.3V or 5V (Set in the operation options in FlyPRO).

  In this mode, programmers will send messages of supply conflicts if they detect the existence of power in target boards. Programmers have the function to detect over-current. In power supply to target boards, current is limited to 250mA, above which, over-current protection occurs. If the target boards' load current exceeds 250mA, please choose "power self-supply by target boards".

- **Power self-supply by target boards**

  In this mode, programmers don't supply power to target boards.

  SP8 programmers support target boards with voltage 2.7V~5.5V. Type driven voltage of ISP adjusts to the VCC voltage of target boards.

Declaration

- In either supply mode, target board's VCC must be connected to the second pin of the ISP interface.
- Keep impending the pins not used, and don't connect them to any information from target boards.
- After programming, all ISP interfaces present high-impedance except VCC and GND.
- In application, only connect any of the GND pins 4, 6, 8, 10 to target board.

Offline program

SP8-F supports offline program. In this mode, programmers don't have to be connected to a computer. Programmers have a built-in 128Mbit data memory. So no extra memory cards. Offline program is easy and efficient. In this mode, programmers automatically detect the placement of chips, start programming, and present results via state lights and a built-in buzzer.

Offline program steps

1. Connect the programmer to a computer through the accessory USB cable, and open FlyPRO to download offline data. For details, please refer to user's guide: Download Offline Data.
2. Disconnect the USB cable, and supply power to the programmer via the accessory adapter. (Must use the adapter, not the USB cable, otherwise it leads to the default online mode, so you cannot start offline operation.)
3. After electrified, the programmer first tests the downloaded offline data, which takes 3~15 seconds. If the test got passed, STA flashes green, indicating that the programmer has entered offline mode. If the test failed, STA flashes red, indicating that the programmer doesn't have effective offline data and thus cannot start offline program.
4. STA flashes green, indicating the programmer is waiting for the chip to be put in.
5. If STA stops flashing and shows orange, the programmer has detected the chip and is programming.
6. When the chip has been programmed, STA shows green for success or red for failure, and the programmer is waiting for that chip to be removed from the socket.
7. After detecting the remove, the programmer repeats steps 4 to 7 to program the next chip.

STA Declaration for offline program

Status of STA	Status Declaration
Flashes green	Waiting for a chip to be put in
Orange	Programming the chip
Green	Succeeded in programming, waiting for the chip to be removed
Red	Failed in programming, waiting for the chip to be removed

Attention

- Chips in ISP mode do not support offline program.
- In offline program, must use the accessory adapter. When connected (and supplied power by) to the computer's USB interface, the programmer can only work in online mode.

Function customization

SP8-F (offline edition) also provides the control signals for automatic program devices. Please call us if you need.

Examine offline data

The offline data downloaded to the programmer can be examined by FlyPRO. For details, please refer to Examine Offline Data in Chapter 4. Software Performance.

Chapter 4. Software Performance

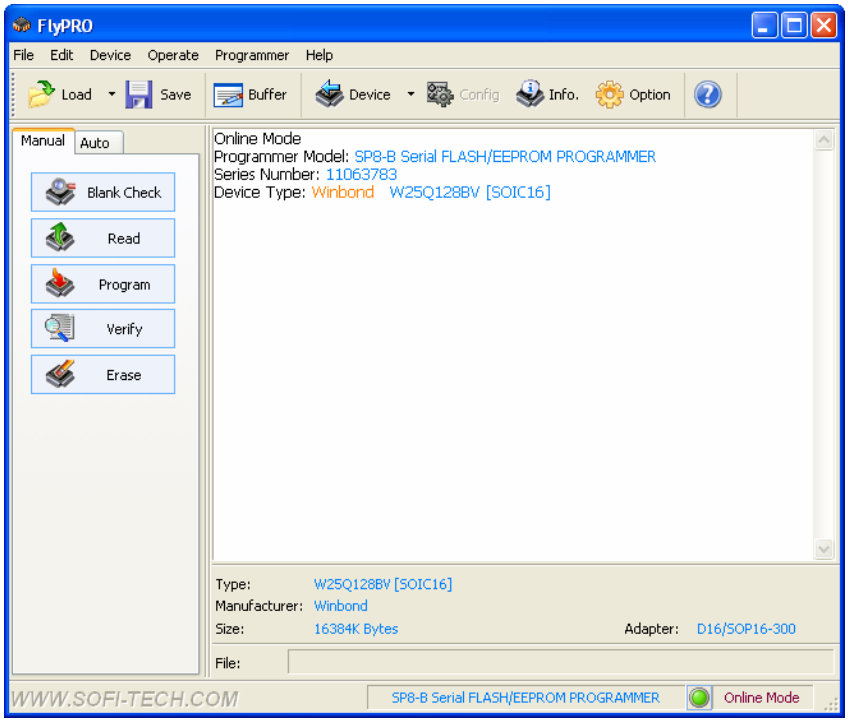
Introduction into software

FlyPRO is the driver software for SOFI programmers and provides all kinds of program operations for chips, such as "Erase", "Read", "Program", "Test" and "Encrypt ". With this software, you can program single chip in socket and ISP download for target boards.

Software properties

- Friendly user interface
- Complete program operations, including erase, vacancy detect, check, program (write), read, read/write fuse address, and encrypt
- Supports automatic program, finishes several operations by one step and automatically sets operation content
- Friendly way of device configuration
- Voice prompts in programming
- Choose to set program areas (for chips with several memory areas)
- Complete buffer compilation supports input and modification by keyboard, and copying, filling and logical operation
- List of recently used devices: You can rapidly change the current device to be the recently used type.
- List of recently loaded files: You can rapidly load the recently loaded files.
- Program counting: Automatically counts the numbers for success and failure
- Examines bad contact of pins and shows results
- Supports mass production mode, automatically examines whether the chip has been put into the socket and starts programming immediately (Some products don't support this function.)
- Automatic serial number: Supports increment mode and user-defined mode (Some products don't support this function.)
- Online help
- Supports WinXP, VISTA, Win7(32bit/64bit)

Main interface

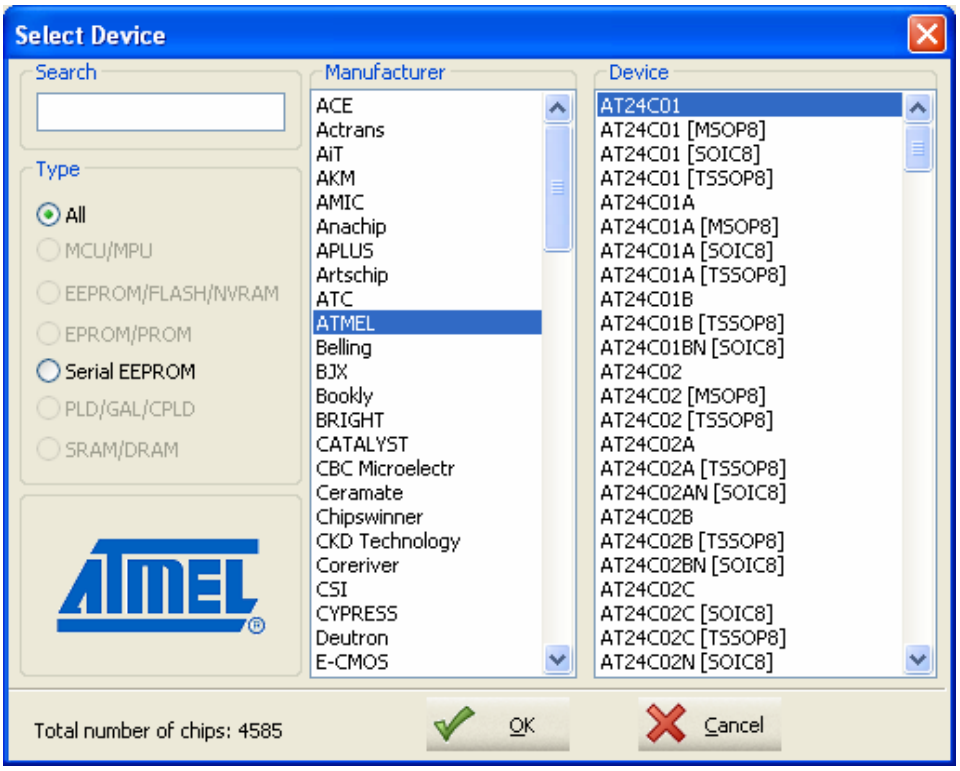


Choose chips

To program, you must choose the right manufacturer and type of chips. Steps to choose chips:

- **Open the dialog Choose Chips**

  Click "Chips" -> "Choose Chips" or corresponding toolbar button, and the following dialog Choose Chips bumps out.



- **Choose chip type**

  Choose the corresponding classification according to the chip type. If not clear about the classification, you can choose "all types".

- **Choose manufacturer**

- **Choose chip model**

  Choose the specific model in the chip box, then click "Yes", or double left click on the corresponding chip in the chip box, and finish this step.

Search for chip model

Enter some words of the chip model in the search box, and the software immediately shows all suited models. For instance, with input of "51", the software shows all models of chips that contain the characters "51".

Declaration of chip models

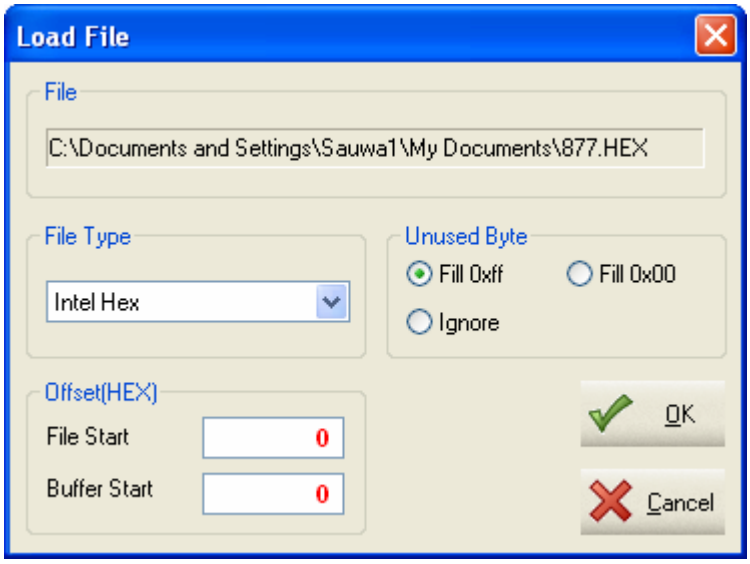
Some chips have the suffix [ ], indicating the encapsulation type, or program mode (For instance, chips programmed in ISP mode have the suffix [ISP]).

You can see the adapter or ISP program connection picture through the chip information window.

Load files

To start programming, you must load appropriate code file to the buffer.

Click [Files] -> [Load files], and the dialog Choose Files bumps out. Choose the appropriate file, and the following setting dialog bumps out.



File types

The software supports many file formats. Include:

Binary

Intel Hex

Motorola S19

TI Tektronix

Extend TI Tektronix

Offset

To load a file, the user can specify start address in the buffer, and the file's start address. For instance, if start address of the file is 0x100, and start address in the buffer is 0x200, data in 0x100 of the file will be loaded to 0x200 of the buffer. Data in 0x101 of the file will be loaded to 0x201 of the buffer. And the following data will be loaded in the same manner.

Attention: Data to the input box are all hex.

Data not used

If byte number of loaded data is less than the size of buffer (for instance, the loaded file AT89S52 with 8K bytes might only use 6K capacity of this chip), this function can set the way to fill in the loaded bytes that are not used.

After data are loaded, the main window presents corresponding message whether the file has been successfully loaded.

If the loaded file exceeds the capacity of the buffer, the software also shows corresponding message for warning. In this case, you must check whether the loaded file is wrong.

Declaration

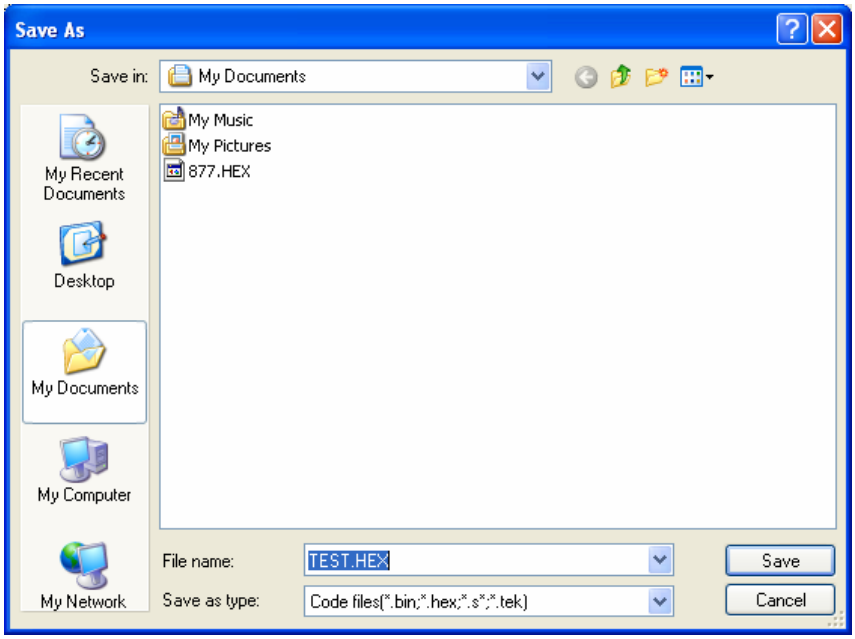
You should choose the right format, otherwise you might fail in loading the file, and thus the chip to be programmed would work abnormally.

After the file is loaded, if you perform the operation to write, the content in the buffer will be covered by the read data. To re-program its content to the chip, you must re-load the file.

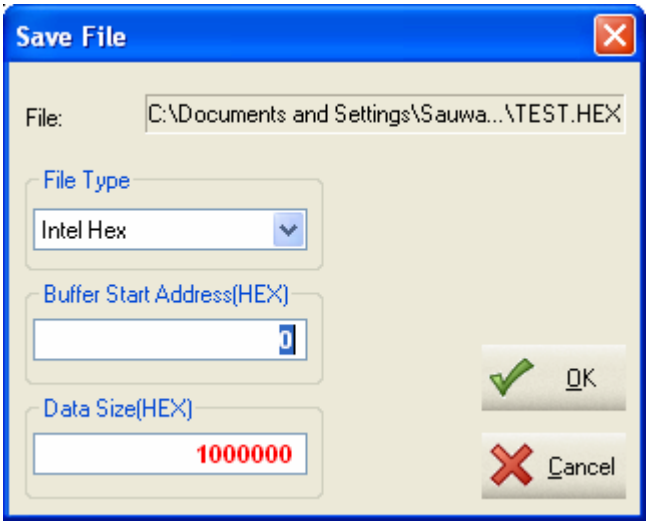
Save files

The data in the buffer can be saved into the hard disk as a file following the steps below:

Click [File] -> [Save file] or the toolbar button "Save", and in the file dialog that bumps out later, choose the path and file name to be saved, as shown below:



Then there will be a setting dialog Save Files. Parameters include file type, start address in the buffer, and data size to be saved.



File type

According to the file's extension, the software automatically chooses a default type for this file. To save in other formats, please choose again.

Start address in the buffer

User can save part of the data in the buffer by setting this option. If the value is not set 0, the data before this address will not be saved to the file.

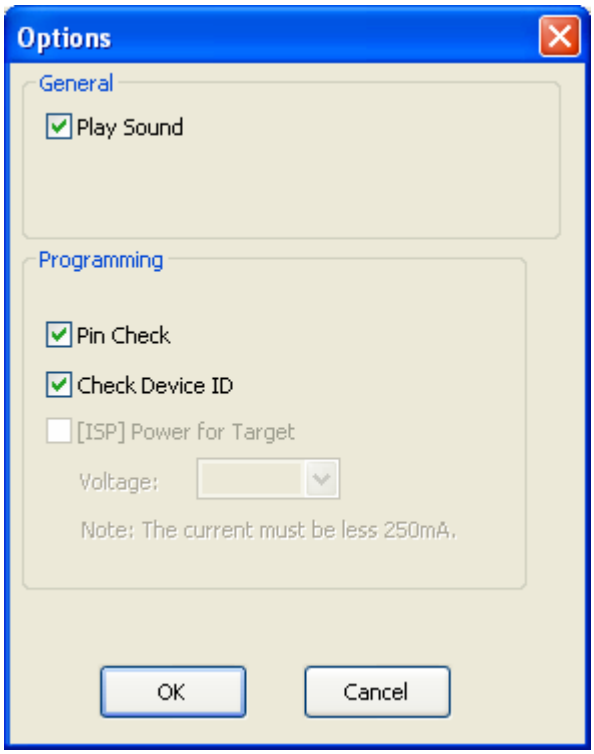
Data size

The number of bytes to be saved. Input format is hex.

Note: All data to be entered must be hex.

Operation options

Click [File] -> [Operation Options] or the toolbar button "Option", and the dialog to set operation options bumps out, as shown below:



Voice prompt

After program is finished, the software automatically plays a sound through the multimedia sound box.
There are two types of sound, one for success and the other for failure.

Examine chip ID

Some chips have an inner ID byte to identify manufacturer and type.
After you start this function, the programmer examines the ID of this chip before performing any operations of program, in order to find out whether the type of the chip chosen matches that of the real programmed chip.

Declaration: Some chips don't have ID identification, and for them, this option is grey-- status of being forbidden.

Detect bad contact of pins

Every time it starts to program, the programmer examines the contact status of the chip's pins, and sends out a corresponding prompt if there is bad contact. Only when all pins are in good contact, will the programmer start programming.

Declaration: ISP mode (chips with the suffix [ISP]) does not support pin detection.

[ISP program] supply power to target boards

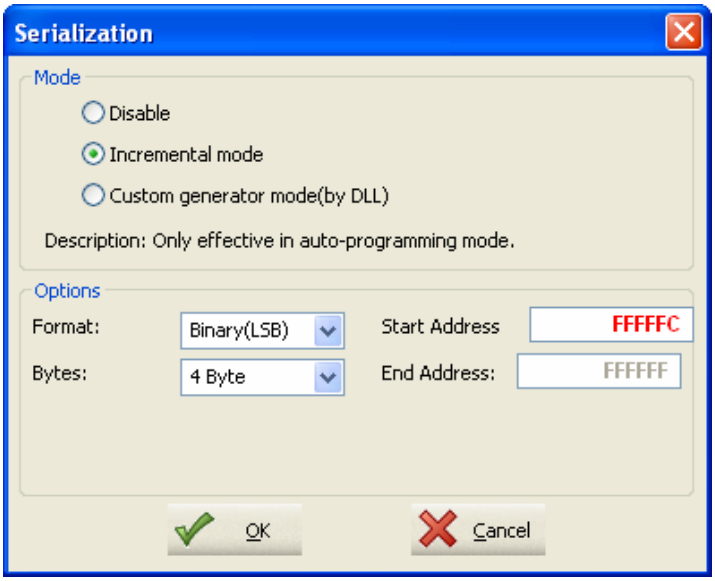
In ISP (chips with suffix [ISP]) program, the programmer can supply power to the target boards (target chips), with a voltage of 3.3V or 5V.
Current limit is 250mA, above which over-current protection occurs.

Automatic Serial Number

Some programmers support automatic serial number, a function used to create exclusive data in a certain area of the target chip.

Set the function and parameters of automatic serial number through [Chips] -> [Automatic Serial Number].

The dialog to set automatic serial number is:



Increment mode

In this mode, once a chip is successfully programmed, the software automatically adds 1 to the number of the appointed area.

Increment mode includes four formats, each with four specified data lengths: 1, 2, 4, 8 bytes.

- **Binary (LSB)**

  In binary format, lower bytes are located at lower address. "Binary" means that data in each byte change in increment from 00H to FFH.

- **Binary (MSB)**

  In binary format, upper bytes are located at lower address. "Binary" means that data in each byte change in increment from 00H to FFH.

- **BCD Code (LSB)**

  In BCD format, lower data are located at lower address. Every byte contains two BCD data-- 4bit represents a decimal number.

  For instance, data in the serial area of the buffer 89 67 45 23 indicate that the current serial number is 23456789. After renewed, the data will be 90 67 45 23.

- **BCD(MSB)**

  In BCD format, upper data are located at lower address. Every byte contains two BCD data-- 4bit represents a decimal number.

  For instance, data in the serial area of the buffer 23 45 67 89 indicate that the current serial number is 23456789. After renewed, the data will be 23 45 67 90.

User mode

After successfully programming a chip, the software transfers exterior user's DLL interface program and passes the pointer of data buffer to the renewed program. User DLL can self-define how to renew data buffer according to needs.

User DLL must define the following two functions:

- **void GetDescription(char * ptext);**

  This function is used to capture the descriptor of this DLL, where ptext is the describing pointer. Descriptor length cannot exceed 128 bytes (64 Chinese characters). Example codes are:

```
const char AUTOSN_DISC[] = "This is a sample。";
/*****************************************************************************
** DLL description                                                          **
** Max: 128 bytes                                                           **
*****************************************************************************/
extern "C" __declspec(dllexport) void GetDescription(char * ptext)
{
    strcpy(ptext，AUTOSN_DISC);
}
```

- **void UpdateSN(BYTE * pBuf，char * pMsg);**

  Function of Renewing Serial Number, which is transferred by the software to renew serial number.

```
/*****************************************************************************
** Function of renewing Automatic Serial Number                             **
** Introduce parameters:
**      pBuf - pointer to a chip's data buffer (not exceeding the chip's capacity)  **
**      pMsg - return description, with up to 64 bytes (32 Chinese characters)      **
**      Every time a chip has been successfully programmed, this information will be **
**      automatically displayed in the operation message window.                    **
*****************************************************************************/
extern "C" __declspec(dllexport) void UpdateSN(BYTE * pBuf，char * pMsg)
{
    DWORD dwCount;

    dwCount = pBuf[0];
    dwCount |= pBuf[1] << 8;
    dwCount |= pBuf[2] << 16;
    dwCount |= pBuf[3] << 24;

    dwCount++;

    pBuf[0] = (BYTE)(dwCount & 0xff);
    pBuf[1] = (BYTE)(dwCount >> 8);
    pBuf[2] = (BYTE)(dwCount >> 16);
    pBuf[3] = (BYTE)(dwCount >> 24);

    sprintf(pMsg，"[SampleSN。DLL] Renewed Automatic Serial Number is: %8。8X"，dwCount);
}
```

After FlyPRO has been installed, there will be, under FlyPro\SampleSN\VC7, an example project code based on the DLL of VS2003(VC70).

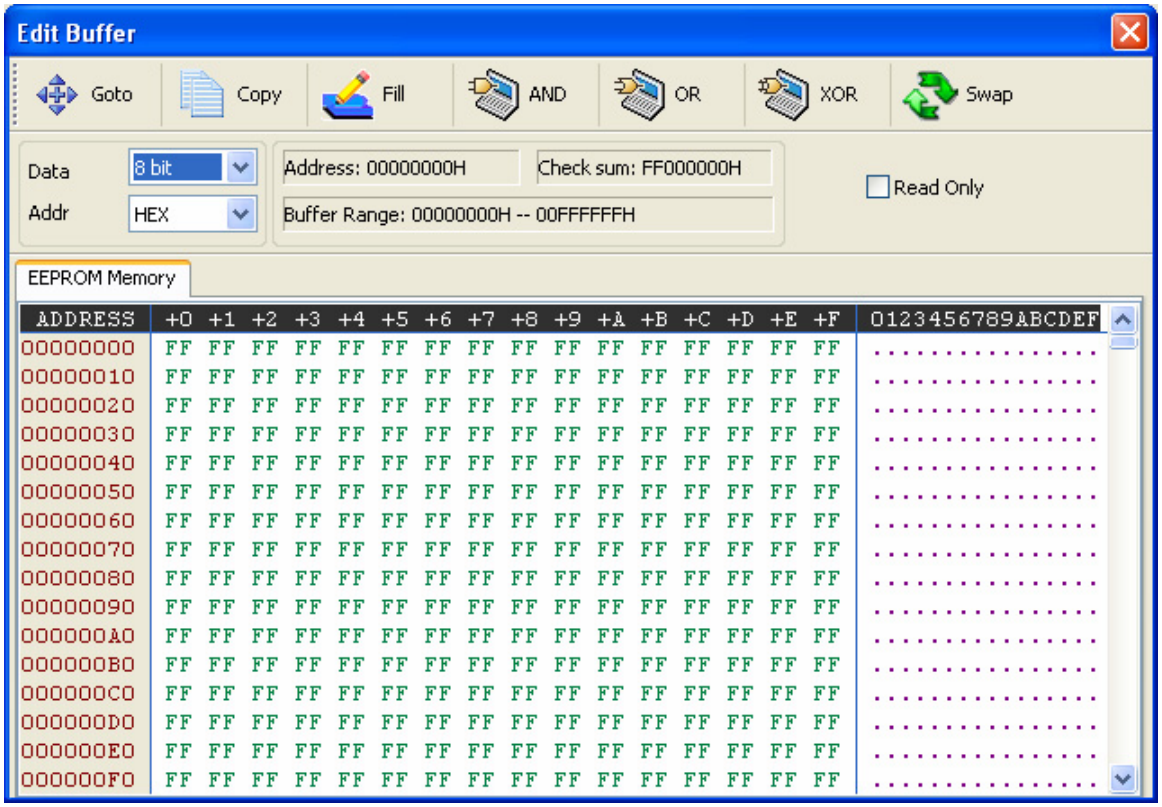
Declaration

- Automatic serial number is effective only in automatic program mode.
- Some programmers don't support the function of automatic serial number.

Buffer compile

The software contains a data buffer, used to save the data loaded from the file or read from the chip. The buffer content can be examined and compiled.

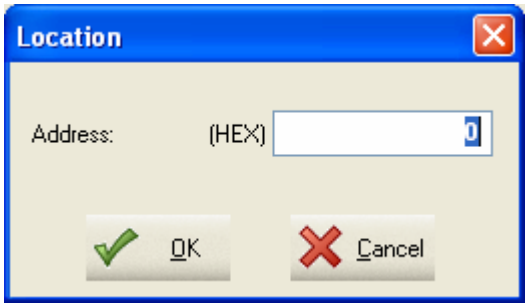
Click [Compile] -> [Data Buffer] to open the dialog Buffer Compile, showed below:



1. Buffer Compile toolbar contains "Set the position of the current cursor" (Position), Copy, Fill, AND/OR/XOR, and exchange between up and low bytes.
2. Set the format of data and address.
3. Show the current status of buffer.
4. Data display area in the buffer - Address column, set as display in decimal or hex.
5. Data display area in the buffer - Area for hex display.
6. Data display area in the buffer – area for ASCII Display.

Position

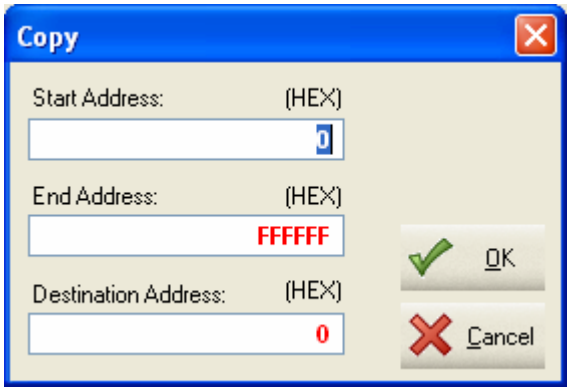
Set the cursor's address, and enter in the dialog that bumps out the address to arrive at, and click "Enter", as showed below:



Attention: Address to be entered is hex.

Copy

Click "Copy", and the following dialog bumps out:

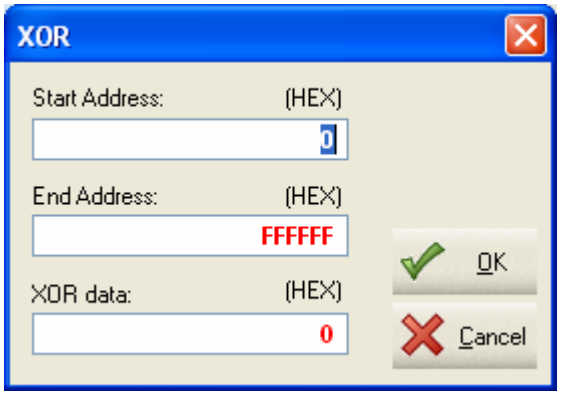


Start address and end address are used to specify source data, and target address is the position the data are to be copied to.

For instance, set start address 0x0, end address 0xFF, target address 0x200, click Enter, and the 256-bit data between 0x0 and 0XFF will be copied to the area between 0x200 and 0x2FF.

AND/OR/XOR

Click corresponding button, and the logic operation dialog bumps out. Take XOR as an example. XOR dialog is:



Start address and end address are used to specify the data area to be conducted with logic operation. XOR data are the logic operation parameters.

For instance, the setting in the left picture means that the 8K data between 0x0 and 0xFFF will be conducted with logic operation with 0x5A.

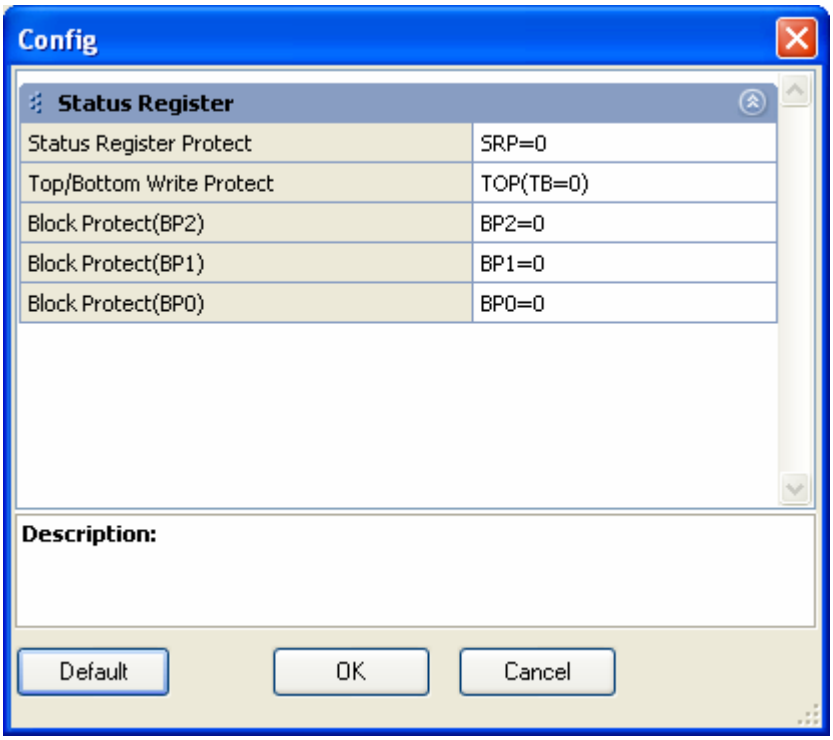
Exchange

In this function, up and low data will be exchanged. That is, data at odd addresses will be exchanged with data at corresponding even addresses. For instance, data at 0x0000 will be exchanged with data at 0x0001, and data at 0x0002 with data at 0x0003, and so on.

Chip configuration

Some chips have corresponding configuration options, and to perform such operations as Fuse location, Configuration word, Encrypt, you must set corresponding chip configuration parameters.

Click [Chip] -> [Chip Configuration], or "configuration options" in the toolbar to open the device configuration dialog.

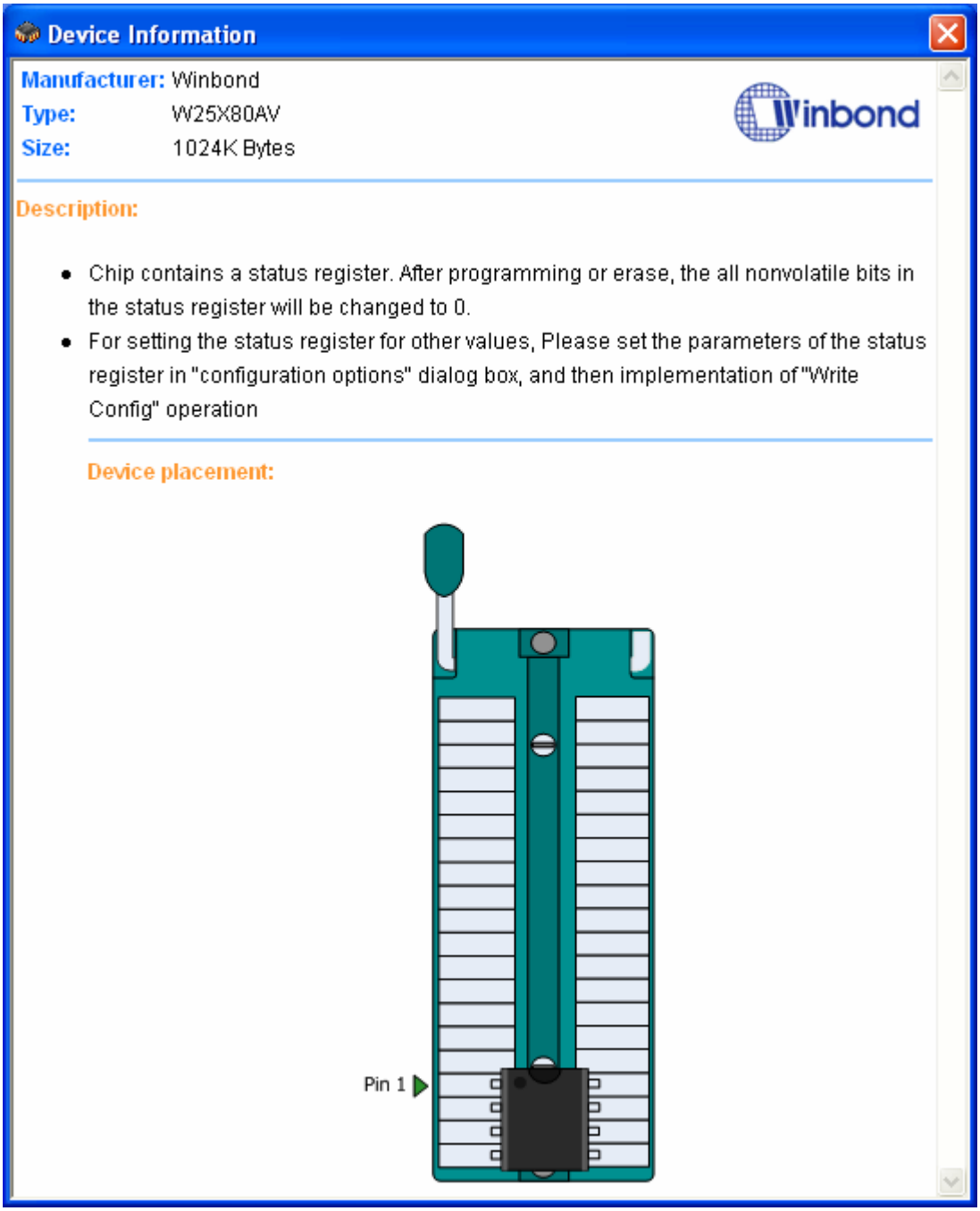


Declaration

- Some chips don't have configuration options.
- Different types of chips have different configuration functions. For details, please refer to corresponding chip manual.

Chip information

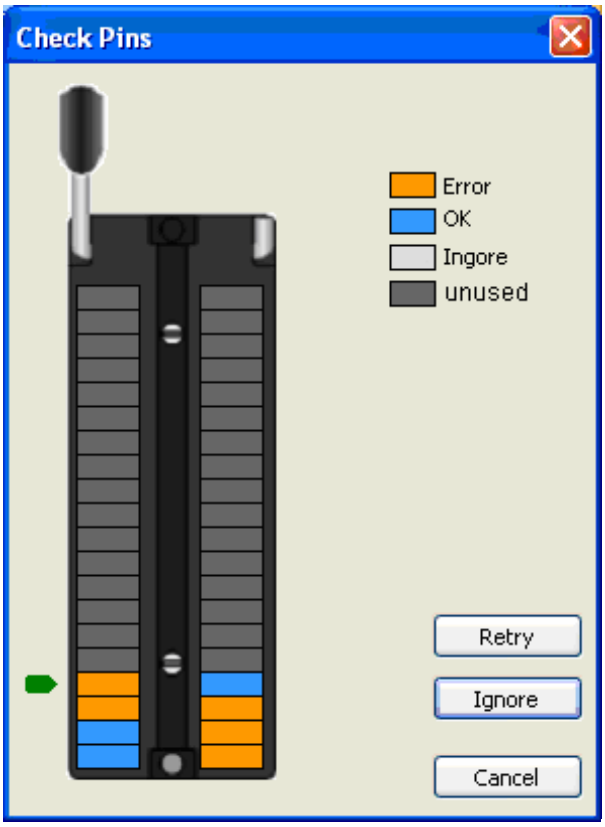
Click "Information" in the toolbar, or menu [Chips] -> [Chip information] to see information of the current chip's adapter, placement, and ISP connection



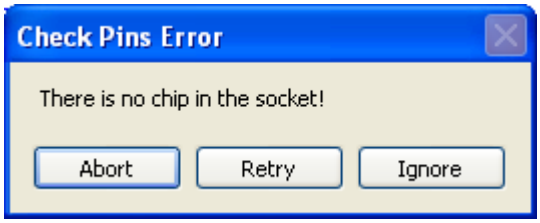
This dialog can be resized through the mouse

Pin detection

The programmer provides status detection of pin contact. When this function is started, it performs pin detection for any chip before programming, and presents the results via figures, as showed below:



When the programmer finds no chip in the socket, the software gives the following prompt:



The user can choose to terminate this operation according to his needs, or try again, or skip pin detection and directly go on to program.

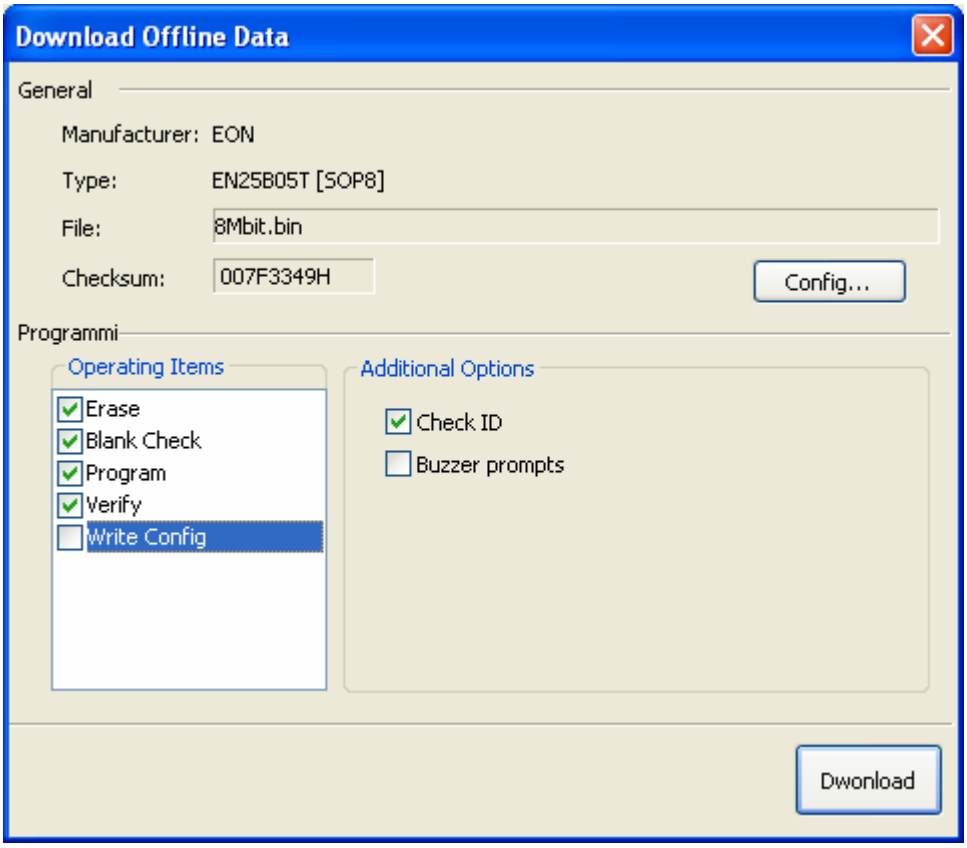
Declaration

- Pin detection can be closed. Refer to operation options for details.

Download offline data

Some programmers don't support offline program, so to perform this function, you must download data needed by offline operation with the help of FlyPRO. Steps to download offline data are:

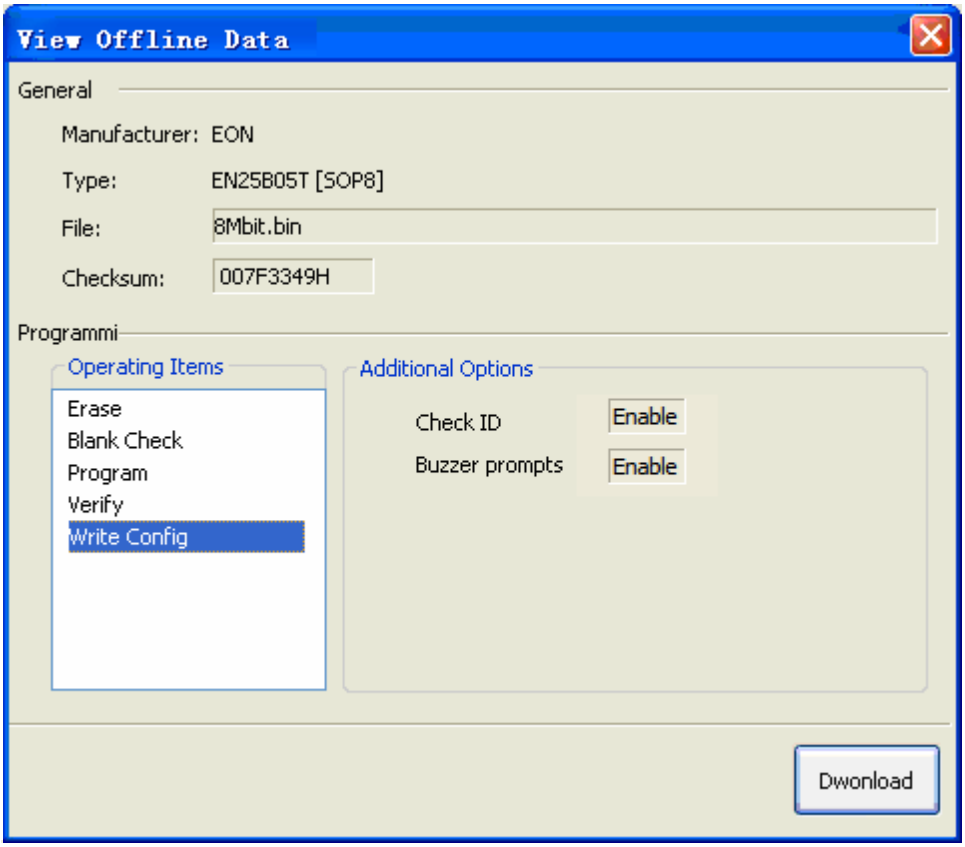
1. Connect the programmer to a computer through the USB cable, start FlyPRO, and connect it to the programmer.
2. Choose the right chip type in FlyPRO, and load the data file to be programmed.
3. Click [Chips] -> [Offline data manage] -> [Download offline data] to open Download Offline Data dialog. The picture is showed below:



4. Set chip configuration options (Such as, if the current chip doesn't have configuration options, the programmer won't display "Chip configuration").
5. Set operation content and accessory options for the chip
6. Click "Download Data" button to download offline data. When it succeeds, the programmer works on its own, without the help from the computer.

Examine offline data

When the programmer is connected with FlyPRO, click [Chips] -> [Offline Data Manage] -> [Examine Offline Data] to see the offline data, as showed below:



No function of upload offline data for the moment

Appendix one, Frequently Asked Questions

☺ Why 24 chips don't have the function Erase?

- These chips are based on EEPROM technology. Users can directly rewrite the data in a chip and don't have to erase in advance. Therefore, there isn't a usable operation of Erase.
- To clear out the data, only write FFH.

☺ Reasons that the software presents chip initialization error

In programming some chips (such as 24 series), the programmer performs initialized checks on them, and if it fails, presents Error. Reasons for chip initialization error are:

- Chip is not put in socket, or pins are in bad contact.
- Misplaced chip, in wrong position or direction
- Built-in problem with the chip
- Chip types don't match-- the type of chip chosen by the software is different from that of the chip put in socket
- Problem with the ISP connecting line (only for ISP, chips with suffix [ISP])

☺ Reasons that the chip to be programmed cannot work normally

Reasons that the chip to be programmed cannot work normally are:

- Failed in loading data file before programming
- Built-in problem with the chip
- Mistakes in the program steps
- Problem with the working circuit or voltage

Appendix two, Common program prompt messages

- **Failed in initializing the chip**

  In programming some chips, the programmer gives such a prompt when no chip in socket, bad contact of the pins, chip placed in wrong direction or built-in problem with the chip.

- **When working current exceeds limit, please check whether type or placement of the chip is wrong.**

  SSP8 programmers have over-current detection system. When the detected current exceeds the limit, the programmer gives a message of error.

  Common reasons for this error are:

  - Chip placed in wrong direction
  - Built-in problem with the chip
  - If the chip uses an adapter, the latter might be in short circuit, or there might be something wrong with its circuit.
  - Chip types don't match-- the type of chip chosen by the software is different from that of chip put in socket

- **Power conflict in ISP, the programmer has detected voltage in target board**

  In ISP program mode, if the current setting is "Programmer supplies to the target board", and the board has current, the programmer presents a message of current conflict.

  Solution: Change the operation option, supply mode changed to target board supply, or close the self-supplied current in the board, and let the programmer supply to the board.

- **The programmer hasn't detected voltage in the target board**

  In ISP program mode, if the current setting is "Self-supply by target board", but the board has not started supply, the programmer presents a message of error.

  Solution: Open the target board's current, or change operation option: supply mode changed to Supply by the Programmer.

- **Warning: the ignored part exceeds the buffer's extent**

  In loading a file, if data amount exceeds the chip's capacity, it presents such a prompt.

  For HEX files, there is no direct relation between file size and data amount. Therefore, it might be that file size is only 1K, but data size exceeds this extent.